

Asleep Alpacas: Justin Zou (Piggy), Wen Hao Dong (Jal Hordan), Yoonah Chang (Yelena), Liesel Wong (King Hagrid)

SoftDev

P00 -- Move Slowly and Fix Things/Creating a website/

2021-10-27

time spent:

Scenario One: Your team has been contracted to create a collaborative storytelling game/website.

- Users will have to register to use the site.
- Logged-in users can either start a new story or add to an existing story.
- When adding to a story,
 - Users are shown only the latest update to the story, not the whole thing.
 - A user can then add some amount of text to the story.
- Once a user has added to a story, they cannot add to it again.
- When creating a new story,
 - Users get to start with any amount of text and give the story a title.
 - Logged in users will be able to read any story they have contributed to on their homepage (the landing page after login).

Database Organization:

stories:

id (INTEGER)	author_id (INTEGER)	title (TEXT)	full_story (TEXT)	last_update (TEXT)
--------------	---------------------	--------------	-------------------	--------------------

users:

id (INTEGER)	username (TEXT)	password (TEXT)
--------------	-----------------	-----------------

contributions:

user_id (INTEGER)	story_id (INTEGER)
-------------------	--------------------

Components and Purpose:

- Templates
 - home.html: Displays a list of the logged in user's stories they contributed to
 - login.html: Contains a form with username and password input
 - register.html: Contains a form with username and password input
 - new.html: Contains a form with the story title and starting body
 - discover.html: Displays a list of stories the user hasn't contributed to
 - view.html: Display a full story and its title

- edit.html: Displays the latest update and has a form containing a textarea tag to continue the story
- Flask App
 - “/”: Renders home.html
 - “/logout”: Pops user_id session if exists and redirects to “/login”
 - “/login”: Handles logging in
 - “/register”: Handles creating new users
 - “/create”: Add new stories
 - “/discover”: Lists stories the user hasn’t contributed to
 - “/story/<story_id>”: Shows the entire story or allows a user to contribute to the story depending on whether or not they’ve already contributed to the story
- SQLite database
 - Relational database containing tables of stories, users, and contributions (a user and which story they’ve contributed to)

Template Representation:

Home Page:

Home	New	Discover	login
			Register

Login Page:

Login

user: _____

pass: _____

Register Page:

Register

Username: _____

Password: _____

Confirm Pass: _____

New Story:

Create Story

Title: _____

Text: _____

Discover:

Discover

<div>Story 1 by: user</div> <div>view edit</div>	<div>Story 2 by: user 2</div> <div>view edit</div>
--	--

View:

Title: _____

by: _____

Text: _____

Edit Page:

Title: _____

by: _____

Last update: _____

Contribution: _____

Linkage: Each of the tabs seen on the homepage will go to their respective page. I.e. New goes to create a story page, discovery goes to discovering page etc. Clicking on a story with the view button from the discovery page will lead to a view page and clicking on the edit page will go to that edit page as well. The User would be able to go back with a navbar like this:

Home	New	Discover	login Register
------	-----	----------	-------------------

The login and register would change to logout if they are logged in already.
The Register, Login, Edit and Create pages will all have a submit button and be submitted through the enter key.

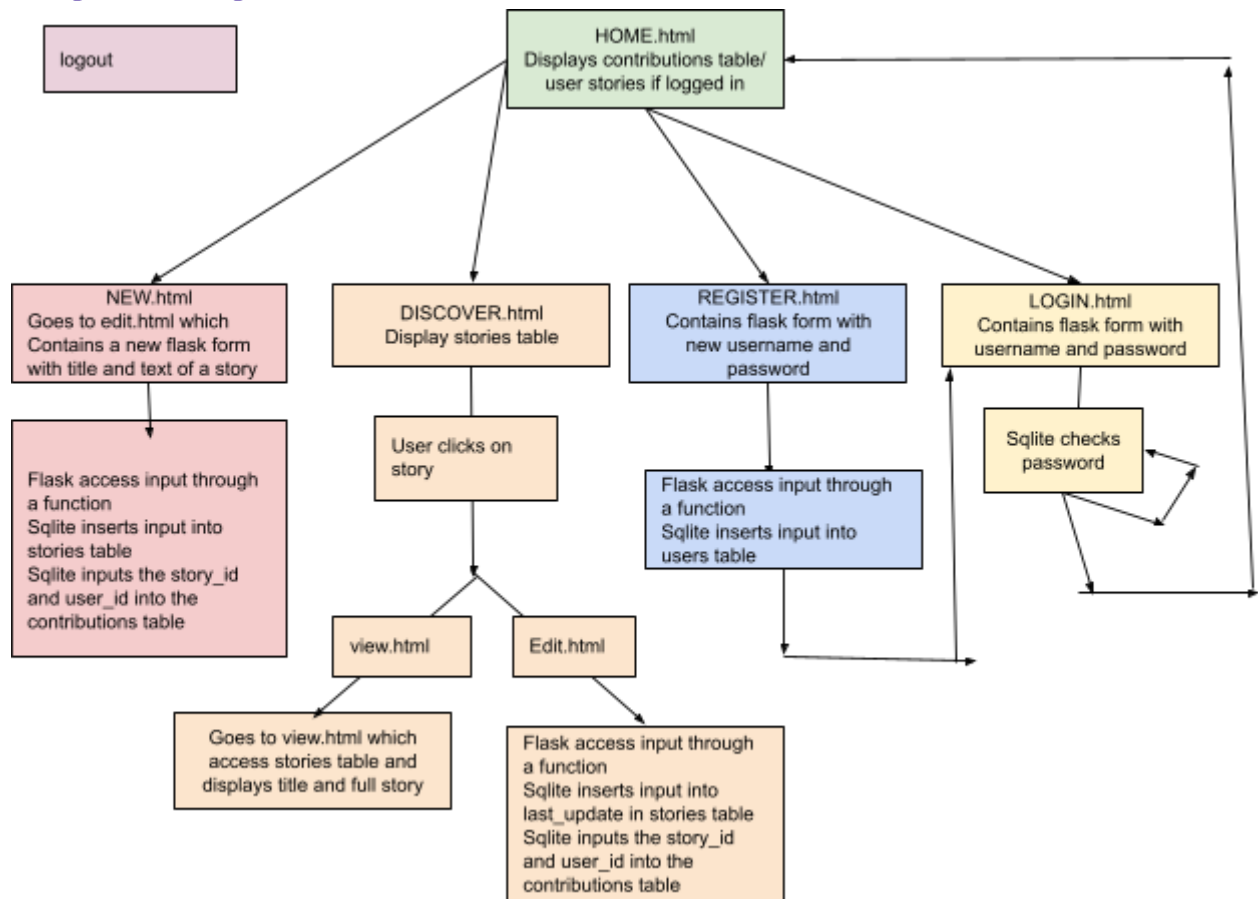
Division of Tasks:

Templates (Front End): Yoonah

SQLite database: Justin

Flask App and Python: Liesel, Wen Hao

Components Map:



The solo Logout Component is to signify that you can logout from any of the pages seen.